PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+process +(allocation +or +allocate +or +allocated +or +alloc SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem

Terms used

process allocation or **allocate** or **allocated** or **allocating sequential** or **consecutive memory concealing** or **co**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced

Try this search i

Results 1 - 20 of 200

Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10 next

1 Register allocation for software pipelined loops

B. R. Rau, M. Lee, P. P. Tirumalai, M. S. Schlansker

July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on design and implementation**, Volume 27 Issue 7

Full text available: pdf(1.84 MB) Additional Information: full citation, abstract, references, citings, in

Software pipelining is an important instruction scheduling technique for efficiently overlapping suc executing them in parallel. This paper studies the task of register allocation for software pipelined hardware features that are specifically aimed at supporting software pipelines. Register allocation certain novel problems leading to unconventional solutions, especially in the presence of hardware

2 External memory algorithms and data structures: dealing with **massive (**

Jeffrey Scott Vitter

June 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 2

Full text available: pdf(828.46 KB) Additional Information: full citation, abstract, references, citings, in

Data sets in large applications are often too massive to fit completely inside the computers interna input/output communication (or I/O) between fast internal memory and slower external memory (performance bottleneck. In this article we survey the state of the art in the design and analysis of algorithms and data structures, where the goal is to exploit locality in order to reduce the I/O cost

Keywords: B-tree, I/O, batched, block, disk, dynamic, extendible hashing, external memory, hie multidimensional access methods, multilevel memory, online, out-of-core, secondary storage, sor

3 The Alpine file system

M. R. Brown, K. N. Kolling, E. A. Taft

November 1985 **ACM Transactions on Computer Systems (TOCS)**, Volume 3 Issue 4

Full text available: pdf(2.95 MB) Additional Information: full citation, abstract, references, citings, in

Alpine is a file system that supports atomic transactions and is designed to operate as a service on primary purpose is to store files that represent databases. An important secondary goal is to store documents, program modules, and the like. Unlike other file servers described in the literature, Al to implement atomic file update. Another unusual aspect of Alpine is that it performs all commu ..

4 Dataflow machine architecture
Arthur H. Veen
December 1986 **ACM Computing Surveys (CSUR)**, Volume 18 Issue 4
Full text available: pdf(3.19 MB) Additional Information: full citation, abstract, references, citings, in

Dataflow machines are programmable computers of which the hardware is optimized for fine-grain computation. The principles and complications of data-driven execution are explained, as well as t fine-grain parallelism. A general model for a dataflow machine is presented and the major design dataflow machines described in the literature are surveyed on the basis of this model and its asso

5 File servers for network-based distributed systems
Liba Svobodova
December 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 4
Full text available: pdf(4.23 MB) Additional Information: full citation, references, citings, index terms

6 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Co**
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index term

Understanding distributed applications is a tedious and difficult task. Visualizations based on proce used to obtain a better understanding of the execution of the application. The visualization tool we developed at the University of Waterloo. However, these diagrams are often very complex and do desired overview of the application. In our experience, such tools display repeated occurrences of

7 System-level power optimization: techniques and tools
Luca Benini, Giovanni de Micheli
April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volu
Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, in

This tutorial surveys design methods for energy-efficient system-level design. We consider electro hardware platform and software layers. We consider the three major constituents of hardware tha computation, communication, and storage units, and we review methods of reducing their energy models for analyzing the energy cost of software, and methods for energy-efficient software desig survery ...

8 Disk cache—miss ratio analysis and design considerations
Alan J. Smith
August 1985 **ACM Transactions on Computer Systems (TOCS)**, Volume 3 Issue 3
Full text available: pdf(3.13 MB) Additional Information: full citation, abstract, references, citings, in

The current trend of computer system technology is toward CPUs with rapidly increasing processin of rapidly increasing density, but with disk performance increasing very slowly if at all. The implica some point the processing power of computer systems will be limited by the throughput of the inp solution to this problem, which is described and evaluated in this paper, is disk cache

9 Storage: Deconstructing storage arrays
Timothy E. Denehy, John Bent, Florentina I. Popovici, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Du
October 2004 **Proceedings of the 11th international conference on Architectural support for and operating systems**
Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, references, index term

We introduce Shear, a user-level software tool that characterizes RAID storage arrays. Shear emp algorithms combined with statistical techniques to automatically determine the important properti the number of disks, chunk size, level of redundancy, and layout scheme. We illustrate the correc upon numerous simulated configurations, and then verify its real-world applicability by running Sh

Keywords: RAID, storage

10 An effective write policy for software coherence schemes
Y.-C. Chen, A. V. Veidenbaum
December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**
Full text available: pdf(1.14 MB) Additional Information: full citation, references, index terms

11 Functional-join processing
R. Braumandl, J. Claussen, A. Kemper, D. Kossmann
February 2000 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volum
Full text available: pdf(486.22 KB) Additional Information: full citation, abstract, index terms

Inter-object references are one of the key concepts of object-relational and object-oriented datab investigate alternative techniques to implement inter-object references and make the best use of in evaluating functional joins. We will give a comprehensive overview and performance evaluation simple (single-valued) as well as multi-valued functional joins. Furthermore, we will describe spec

Keywords: *Functional join, Logical OID, Object identifier, Order-preserving join, Physical OID, Po*

12 Cache coherence in large-scale shared-memory multiprocessors: issues and comparisons
David J. Lilja
September 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 3
Full text available: pdf(3.12 MB) Additional Information: full citation, references, citings, index terms

13 Revising old friends: Capriccio: scalable threads for internet services
Rob von Behren, Jeremy Condit, Feng Zhou, George C. Necula, Eric Brewer
October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems principl**
Full text available: pdf(312.83 KB) Additional Information: full citation, abstract, references, index term

This paper presents Capriccio, a scalable thread package for use with high-concurrency servers. W event-based systems, we believe that thread-based systems can provide a simpler programming or superior performance.By implementing Capriccio as a user-level thread package, we have deco implementation from the underlying operating system. As a result, we can take advantage of coop asynchronous ...

Keywords: blocking graph, dynamic stack growth, linked stack management, resource-aware sch

14 Use-Based Register Caching with Decoupled Indexing
March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual i Computer architecture**, Volume 32 Issue 2
Full text available: pdf(182.25 KB) Additional Information: full citation, abstract

Wide, deep pipelines need many physical registersto hold the results of in-flight instructions. Simu frequencies prohibit using largeregister files and bypass networks without a significantperformanc techniquesusing register caching to reduce this penalty sufferfrom several problems including poo decisions and the need for a fully-associativecache for good performance. We present novelmecha indexin ...

15 Towards a theory of cost management for digital libraries and electronic commerce
A. Prasad Sistla, Ouri Wolfson, Yelena Yesha, Robert Sloan
December 1998 **ACM Transactions on Database Systems (TODS)**, Volume 23 Issue 4
Full text available: pdf(246.67 KB) Additional Information: full citation, abstract, references, citings, in

One of the features that distinguishes digital libraries from traditional databases is new cost mode intellectual property. Clients will pay for accessing data items in digital libraries, and we believe th be as important as optimizing performance in traditional databases. In this article we discuss cost accessing digital libraries, with the objective of determining the minimum cost protocol for each m

Keywords: average case analysis, caching, cost models, demand, on-line services, protocols, sub

16 Mapping irregular applications to DIVA, a PIM-based data-intensive architecture
Mary Hall, Peter Kogge, Jeff Koller, Pedro Diniz, Jacqueline Chame, Jeff Draper, Jeff LaCoss, John Gra Srivastava, William Athas, Vincent Freeh, Jaewook Shin, Joonseok Park
January 1999 **Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(111.41 KB) Additional Information: full citation, references, citings, index terms

17 HiPE on AMD64
Daniel Luna, Mikael Pettersson, Konstantinos Sagonas
September 2004 **Proceedings of the 2004 ACM SIGPLAN workshop on Erlang**
Full text available: pdf(245.37 KB) Additional Information: full citation, abstract, references, index term

Erlang is a concurrent functional language designed for developing large-scale, distributed, fault-t implementation of the language is the Erlang/OTP system from Ericsson. Even though Erlang/OTP machine interpreter, it nowadays also includes the HiPE (High Performance Erlang) native code co component. This paper describes the recently developed port of HiPE to the AMD64 architecture. W

Keywords: AMD64, erlang, native code compilation

18 A specification of JOVIAL
Christopher J. Shaw
December 1963 **Communications of the ACM**, Volume 6 Issue 12
Full text available: pdf(1.93 MB) Additional Information: full citation, references, citings

19 Database concurrency control using data flow graphs
M. H. Eich, David L. Wells
June 1988 **ACM Transactions on Database Systems (TODS)**, Volume 13 Issue 2
Full text available: pdf(2.42 MB) Additional Information: full citation, abstract, references, citings, in

A specialized data flow graph, Database Flow Graph (DBFG) is introduced. DBFGs may be used for operations, particularly in an MIMD database machine environment. A DBFG explicitly maintains in intratransaction dependencies, and is constructed from the Transaction Flow Graphs (TFG) of activ is the generalization of a query tree used, for example, in DIRECT [15]. All DBFG schedules ...

20 HFS: a performance-oriented flexible file system based on building-block compositions
Orran Krieger, Michael Stumm
August 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 3

Full text available: pdf(383.87 KB) Additional Information: full citation, abstract, references, citings, in

The Hurricane File System (HFS) is designed for (potentially large-scale) shared-memory multipro based on the principle that, in order to maximize performance for applications with diverse require support a wide variety of file structures, file system policies, and I/O interfaces. Files in HFS are im building blocks composed in potentially complex ways. This approach yields great flexibility, allow

Keywords: customization, data partitioning, data replication, flexibility, parallel computing, paral

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 nex

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player R